**闭包概念：**

闭包指的是：能够访问另一个函数作用域的变量的函数。

**闭包demo：**

function A(){

function B(){

console.log('Hello Closure!');

}

return B;

}

var C = A();

C();// Hello Closure!

**为什么使用闭包？**

Javascript 中的 GC 机制:

在 Javascript 中，如果一个对象不再被引用，那么这个对象就会被 GC 回收，否则这个对象一直会保存在内存中。

在上述例子中，B 定义在 A 中，因此 B 依赖于 A ,而外部变量 C 又引用了 B , 所以A间接的被 C 引用。

也就是说，A 不会被 GC 回收，会一直保存在内存中。

**改进例子：**

function A() {

var count = 0;

function B() {

count ++;

console.log(count);

}

return B;

}

var C = A();

C();// 1

C();// 2

C();// 3

**分析：**count 是函数A 中的一个变量，它的值在函数B 中被改变，函数 B 每执行一次，count 的值就在原来的基础上累加 1 。因此，函数A中的 count 变量会一直保存在内存中。

当我们需要在模块中定义一些变量，并希望这些变量一直保存在内存中但又不会 “污染” 全局的变量时，就可以用闭包来定义这个模块。